

```

1  /*
2  TinyTrackGPS.cpp - A simple track GPS to SD card logger.
3  TinyTrackGPS v0.8
4
5  Copyright © 2019-2021 Francisco Rafael Reyes Carmona.
6  All rights reserved.
7
8  rafael.reyes.carmona@gmail.com
9
10 This file is part of TinyTrackGPS.
11
12 TinyTrackGPS is free software: you can redistribute it and/or modify
13 it under the terms of the GNU General Public License as published by
14 the Free Software Foundation, either version 3 of the License, or
15 (at your option) any later version.
16
17 TinyTrackGPS is distributed in the hope that it will be useful,
18 but WITHOUT ANY WARRANTY; without even the implied warranty of
19 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
20 GNU General Public License for more details.
21
22 You should have received a copy of the GNU General Public License
23 along with TinyTrackGPS. If not, see <https://www.gnu.org/licenses/>.
24 */
25
26 /*****
27 / Programa de localizacion por gps que graba las posiciones en
28 / un fichero de texto cada segundo, de forma diaria.
29 /
30 / - Conectar módulo SD con pin CS (naranja) en pin 10 arduino.
31 /
32 / Uso de librería TinyGPS.
33 / Requiere uso de librería SoftwareSerial, se presupone que disponemos
34 / de un dispositivo GPS serie de 9600-bauds conectado en pines 9(rx) y 8(tx).
35 / - Conectar módulo NMEA-6M (gps) pines 8,9 (9 - pin rx negro)
36 /
37 / - Conectar LCD 16x2 pines 2,3,4,5,6,7 (2-amarillo , 3-azul,
38 / 4-rojo, 5-azul oscuro, 6-verde, 7-blanco)
39 /
40 / - Conectar OLED 0.96" en SDA y SCL. pines A4 y A5 del Arduino UNO.
41 *****/
42
43 // Include libraries.
44 #include <Arduino.h>
45 #include "config.h"
46 #include "Display.h"
47 #include <SoftwareSerial.h>
48 #include <TinyGPS.h>
49 #include <SdFat.h>
50 #include <sdios.h>
51 #include <LowPower.h>
52 #include <UTMConversion.h>
53
54 // Variables para grabar en SD.
55 char GPSLogFile[] = "YYYYMMDD.csv"; // Formato de nombre de fichero. YYYY-Año, MM-
Mes, DD-Día.
56
57 const uint8_t CHIP_SELECT = SS; // SD card chip select pin. (10)
58 SdFat card; //SdFat.h library.
59 SdFile file;

```

```

60 boolean SDReady;
61 boolean SaveOK;
62
63 // Variables y clases para obtener datos del GPS y conversion UTM.
64 TinyGPS gps;
65 GPS_UTM utm;
66 SoftwareSerial gps_serial(9, 8);
67 int year_actual;
68 byte month_actual, day_actual;
69 byte hour_prev, minute_prev, second_prev;
70 float flat, flon;
71 int year;
72 byte month, day, hour, minute, second, hundredths;
73 unsigned long age;
74 unsigned int elev;
75
76 // Definimos el Display
77 #if defined(DISPLAY_TYPE_LCD_16X2)
78 Display LCD(LCD_16X2);
79 #elif defined(DISPLAY_TYPE_LCD_16X2_I2C)
80 Display LCD(LCD_16X2_I2C);
81 #elif defined(DISPLAY_TYPE_SDD1306_128X64)
82 Display LCD(SDD1306_128X64);
83 #else
84 #define NO_DISPLAY
85 #endif
86
87 //-----
88 /*
89  * User provided date time callback function.
90  * See SdFile::dateTimeCallback() for usage.
91  */
92 void dateTime(uint16_t* date, uint16_t* time) {
93     // User gets date and time from GPS or real-time
94     // clock in real callback function
95
96     // return date using FAT_DATE macro to format fields
97     /*date = FAT_DATE(year, month, day);
98     *date = (year-1980) << 9 | month << 5 | day;
99
100    // return time using FAT_TIME macro to format fields
101    /*time = FAT_TIME(hour, minute, second);
102    *time = hour << 11 | minute << 5 | second >> 1;
103 }
104 //-----
105
106 void GPSData(TinyGPS &gps, GPS_UTM &utm);
107 #ifndef NO_DISPLAY
108 void ScreenPrint(Display &LCD, TinyGPS &gps, GPS_UTM &utm);
109 bool pinswitch();
110 #endif
111 void GPSRefresh();
112
113 unsigned long iteration = 0;
114
115 void setup(void) {
116     //Serial.begin(9600);
117     gps_serial.begin(9600);
118
119     //Serial.print(F("Initializing SD card..."));

```

```

120
121 SDReady = card.begin(CHIP_SELECT);
122 //(SDReady) ? Serial.println(F("Done.)) : Serial.println(F("FAILED!"));
123
124 /* Iniciaizaci3n del display LCD u OLED */
125 #ifndef NO_DISPLAY
126 LCD.start();
127 LCD.clr();
128 LCD.splash(750);      // Dibujamos la presensaci3n.
129 #endif
130
131 //Serial.print(F("Waiting for GPS signal..."));
132 #ifndef NO_DISPLAY
133 LCD.clr();
134 LCD.print("Waiting for","GPS signal...","");
135 unsigned int time = 0;
136 #endif
137
138 bool config = false;
139
140 do {
141     #ifndef NO_DISPLAY
142     LCD.wait_anin(time++);
143     #endif
144     for (unsigned long start = millis(); millis() - start < 1000;) {
145         while (gps_serial.available() > 0) {
146             char c = gps_serial.read();
147             //Serial.write(c); // uncomment this line if you want to see the GPS data
flowing
148             if (gps.encode(c)) {// Did a new valid sentence come in?
149                 gps.crack_datetime(&year, &month, &day, &hour, &minute, &second,
&hundredths, &age);
150                 (age != TinyGPS::GPS_INVALID_AGE) ? config = true : config = false;
151             }
152         }
153     }
154 }while(!config);
155
156 sprintf(GPSLogFile, "%04d%02d%02d.csv", year, month, day);
157
158 year_actual = year;
159 month_actual = month;
160 day_actual = day;
161 hour_prev = hour;
162 minute_prev = minute;
163 second_prev = second;
164
165 //Serial.println(F("Done.));
166 //Serial.println(F("Configuration ended.));
167
168 #ifndef NO_DISPLAY
169 LCD.clr();
170 #endif
171 }
172
173 void loop(void) {
174     bool gps_ok = false;
175
176     while (gps_serial.available() > 0) {
177         char c = gps_serial.read();

```

```

178     //Serial.write(c); // uncomment this line if you want to see the GPS data
    flowing
179     if (gps.encode(c)) {
180         gps_ok = true;
181     }
182 }
183
184 gps.f_get_position(&flat, &flon, &age);
185 utm.UTM(flat, flon);
186 gps.crack_datetime(&year, &month, &day, &hour, &minute, &second, &hundredths,
    &age);
187 if ((elev = gps.altitude()) == TinyGPS::GPS_INVALID_ALTITUDE) elev = 0;
188 else elev /= 100L;
189
190 if (gps_ok) {
191     GPSRefresh();
192     if (!((hour_prev == hour) && (minute_prev == minute) && (second_prev ==
    second))) {
193         GPSData(gps, utm);
194         iteration++;
195     }
196     #ifndef NO_DISPLAY
197     ScreenPrint(LCD, gps, utm);
198     #endif
199 }
200
201 LowPower.idle(SLEEP_120MS, ADC_OFF, TIMER2_OFF, TIMER1_OFF, TIMER0_OFF, SPI_ON,
    USART0_ON, TWI_ON);
202 }
203
204 void GPSData(TinyGPS &gps, GPS_UTM &utm) {
205     char buffer[60];
206     char line[11];
207     int index;
208     int zone;
209     char band;
210     long X;
211     long Y;
212
213     zone = utm.zone();
214     band = utm.band();
215     X = utm.X();
216     Y = utm.Y();
217
218     if (age != TinyGPS::GPS_INVALID_AGE){
219         index = snprintf(buffer,10, "%02d:%02d:%02d,", hour, minute, second);
220         dtostrf(flat, 10, 6, line);
221         index += snprintf(buffer+index,12,"%s,",line);
222         dtostrf(flon, 10, 6, line);
223         index += snprintf(buffer+index,12,"%s,",line);
224         index += snprintf(buffer+index,7,"%05u,",elev);
225         index += snprintf(buffer+index,19,"%02d%c %ld %ld", zone, band, X, Y);
226         //Serial.print(buffer);
227     }
228
229     if (year != year_actual || month != month_actual || day != day_actual) {
230         sprintf(GPSLogFile, "%04d%02d%02d.csv", year, month, day);
231         year_actual = year;
232         month_actual = month;
233         day_actual = day;

```

```

234 }
235
236 SdFile::dateTimeCallback(dateTime);
237
238 // Si no existe el fichero lo crea y añade las cabeceras.
239 if (SDReady && !card.exists(GPSLogFile)) {
240     if (file.open(GPSLogFile, O_CREAT | O_APPEND | O_WRITE)) {
241         //Serial.print(F("New GPSLogFile, adding heads..."));
242         file.println(F("Time, Latitude, Longitude, Elevation, UTM Coords (WGS84)"));
243         //Serial.println(F("Done.));
244         file.close();
245     }
246     //else {
247         //Serial.println(F("*** Error creating GPSLogFile. ***));
248     //}
249 }
250
251 //if (!((hour_prev == hour) && (minute_prev == minute) && (second_prev ==
second))) {
252     if (SDReady && file.open(GPSLogFile, O_APPEND | O_WRITE)) {
253         //Serial.print(F("Open GPSLogFile to write...));
254         SaveOK = true;
255         file.println(buffer);
256         file.close();
257         //Serial.println(F("Done.));
258         hour_prev = hour;
259         minute_prev = minute;
260         second_prev = second;
261     } else {
262         SaveOK = false;
263         //Serial.println(F("*** Error opening GPSLogFile. ***));
264     }
265     //} //else Serial.println(F("*** GPS signal lost. ***));
266 }
267
268 #ifndef NO_DISPLAY
269 void ScreenPrint(Display &LCD, TinyGPS &gps, GPS_UTM &utm){
270     bool print_utm = false;
271     bool print_grades = false;
272     unsigned short sats;
273
274     sats = gps.satellites();
275
276     if (LCD.display_type() == SDD1306_128X64) {
277         print_utm = true;
278         print_grades = true;
279     }
280     else if (!pinswitch()) print_utm = true;
281     else print_grades = true;
282
283     if (print_utm) {
284         char line[12];
285
286         sprintf(line, "%02d%c %ld ", utm.zone(), utm.band(), utm.X());
287         //Serial.println(line);
288         LCD.print(0,0,line);
289         LCD.print_PChar((byte)6);
290         sprintf(line, "%02hu ", sats);
291         //Serial.println(line);
292         LCD.print(12,0,line);

```

```

293     SaveOK ? LCD.print_PChar((byte)7) : LCD.print("-");
294
295     // New line
296     sprintf(line, "%ld ", utm.Y());
297     //Serial.println(line);
298     LCD.print(1,1,line);
299     LCD.print_PChar((byte)5);
300     LCD.print(10,1,"_____");
301     sprintf(line, "%um", elev);
302     //Serial.println(line);
303
304     if (elev < 10) LCD.print(14,1,line);
305     else if (elev < 100) LCD.print(13,1,line);
306     else if (elev < 1000) LCD.print(12,1,line);
307     else LCD.print(11,1,line);
308 }
309
310 if (print_grades) {
311     char line[11];
312     LCD.print(1,(LCD.display_type() == SDD1306_128X64) ? 2 : 0,"LAT=");
313     dtostrf(flat, 10, 6, line);
314     LCD.print(line);
315     LCD.print(1,(LCD.display_type() == SDD1306_128X64) ? 3 : 1,"LON=");
316     dtostrf(flon, 10, 6, line);
317     LCD.print(line);
318 }
319 }
320
321 bool pinswitch()
322 {
323     bool pin;
324
325     if (LCD.display_type() == SDD1306_128X64) return true;
326
327     pin = bitRead(iteration,4); // Change every 8 seconds.
328     //LCD.clr(); -> Too slow clear individual characters.
329     if ((iteration%16) == 0) {
330         LCD.print(0,0," ");
331         LCD.print(15,0," ");
332         LCD.print(0,1," ");
333         LCD.print(15,1," ");
334     }
335     return pin;
336 }
337 #endif
338
339 void GPSRefresh()
340 {
341     while (gps_serial.available() > 0)
342         gps.encode(gps_serial.read());
343 }
344

```